# Arctor: Experimental Evaluation

We have evaluated the *Arctor* termination prover (`http://www.react.uni-saarland.de/tools/arctor/`) on a number of multi-threaded benchmarks, and compared it to the termination provers Terminator, T2, and AProVE.

## 1 Simple Benchmarks

These benchmarks are quite simple multi-threaded programs, intended to compare how well different techniques can handle concurrency. Typical thread here consists just of several instructions. Table 1 shows the performance of the termination provers on the benchmarks, described below. The archive with this set of benchmarks is available at `http://www.react.uni-saarland.de/tools/arctor/simple-benchmarks.zip`.

**Producer-Consumer.** The *Producer-Consumer* benchmark is a simplified model of the *map-reduce* architecture from distributed processing: producers model the mapping step for separate data sources, consumers model the reducing step for different types of input data. The natural requirement for such an architecture is that the distributed processing terminates for any finite amount of input data.

**Chain.** The *Chain* benchmark consists of a chain of $n$ threads, where each thread decreases its own counter $x_i$, but the next thread in the chain can counteract, and increase the counter of the previous thread. Only the last thread in the chain terminates unconditionally.

**Phase.** The *Phase* benchmark is similar to the Chain benchmark, except that now each thread can either increase or decrease its counter $x_i$. Each such *phase change* is, however, guarded by the next thread in the chain, which limits the number of times the phase change can occur.

**Semaphore.** The *Semaphore* benchmark represents a model of a concurrent system where access to a critical resource is guarded by semaphores. Each thread enters its critical section by decreasing the semaphore value if possible, or waits for another thread to leave the critical section and increase the semaphore. We verify *individual accessibility* for a particular thread (i.e., the system without the thread should terminate) under the assumption of a *fair scheduler*.

## 2 Industrial Benchmarks

These benchmarks represent abstractions of real-life industrial multi-threaded programs. Typical thread here consists of dozens of instructions.

In our experiments we have found out that only Arctor can handle these benchmarks. All other termination provers reached either a memout, or an internal timeout, or returned "termination unknown". Therefore, in Table 2 we show

| | Terminator | | T2 | | AProVE | | Arctor | | |
|---|---|---|---|---|---|---|---|---|---|
| Threads | Time(s) | Mem.(MB) | Time(s) | Mem.(MB) | Time(s) | Mem.(MB) | Time(s) | Mem.(MB) | Vertices |
| Chain 2 | 0.65 | 20 | 0.52 | 20 | 1.58 | 131 | 0.002 | 2.0 | 3 |
| Chain 4 | 1.45 | 25 | 0.54 | 22 | 2.13 | 153 | 0.002 | 2.2 | 7 |
| Chain 6 | 24.4 | 57 | 0.58 | 24 | 2.58 | 171 | 0.002 | 2.5 | 11 |
| Chain 8 | × | MO | 0.63 | 26 | 3.48 | 210 | 0.002 | 2.5 | 15 |
| Chain 20 | × | MO | 2.36 | 55 | 16.5 | 941 | 0.007 | 2.5 | 39 |
| Chain 40 | × | MO | 40.5 | 288 | 536 | 1237 | 0.023 | 2.8 | 79 |
| Chain 60 | × | MO | Z3-TO | × | × | MO | 0.063 | 3.0 | 119 |
| Chain 80 | × | MO | Z3-TO | × | × | MO | 0.145 | 3.3 | 159 |
| Chain 100 | × | MO | Z3-TO | × | × | MO | 0.320 | 3.9 | 199 |
| Phase 1 | × | MO | U(4.53) | 48 | 1.60 | 132 | 0.002 | 2.4 | 2 |
| Phase 2 | × | MO | U(4.53) | 48 | 2.16 | 144 | 0.002 | 2.4 | 11 |
| Phase 3 | × | MO | U(30.6) | 301 | 3.83 | 199 | 0.002 | 2.5 | 20 |
| Phase 4 | × | MO | × | MO | 8.89 | 336 | 0.003 | 2.6 | 29 |
| Phase 8 | × | MO | × | MO | 47.0 | 1506 | 0.003 | 2.6 | 65 |
| Phase 10 | × | MO | × | MO | × | MO | 0.012 | 2.7 | 83 |
| Phase 20 | × | MO | × | MO | × | MO | 0.061 | 3.3 | 173 |
| Phase 40 | × | MO | × | MO | × | MO | 0.35 | 4.0 | 353 |
| Phase 60 | × | MO | × | MO | × | MO | 1.18 | 4.2 | 533 |
| Phase 80 | × | MO | × | MO | × | MO | 3.21 | 5.1 | 713 |
| Phase 100 | × | MO | × | MO | × | MO | 7.38 | 6.1 | 893 |
| Semaphore 1 | 3.05 | 26 | 2.81 | 46 | 3.22 | 230 | 0.002 | 2.6 | 8 |
| Semaphore 2 | 622 | 691 | U(20.7) | 219 | U(6.52) | 465 | 0.002 | 2.6 | 16 |
| Semaphore 3 | × | MO | U(15.8) | 239 | U(10.42) | 1138 | 0.003 | 2.6 | 24 |
| Semaphore 10 | × | MO | U(83.5) | 470 | U(246) | 1287 | 0.023 | 2.8 | 80 |
| Semaphore 20 | × | MO | × | MO | × | MO | 0.073 | 3.3 | 160 |
| Semaphore 40 | × | MO | × | MO | × | MO | 0.264 | 4.0 | 320 |
| Semaphore 60 | × | MO | × | MO | × | MO | 0.58 | 4.0 | 480 |
| Semaphore 80 | × | MO | × | MO | × | MO | 1.02 | 4.6 | 640 |
| Semaphore 100 | × | MO | × | MO | × | MO | 1.59 | 5.1 | 800 |
| Producer 1 | 3.37 | 26 | 2.42 | 38 | 3.17 | 237 | 0.002 | 2.3 | 6 |
| Producer 2 | 1397 | 1394 | 3.25 | 44 | 6.79 | 523 | 0.002 | 2.6 | 11 |
| Producer 3 | × | MO | U(29.2) | 253 | U(26.6) | 1439 | 0.002 | 2.6 | 21 |
| Producer 4 | × | MO | U(36.6) | 316 | U(71.2) | 1455 | 0.003 | 2.7 | 30 |
| Producer 5 | × | MO | U(30.7) | 400 | U(312) | 1536 | 0.007 | 2.7 | 44 |
| Producer 10 | × | MO | Z3-TO | × | × | MO | 0.027 | 3.0 | 135 |
| Producer 20 | × | MO | Z3-TO | × | × | MO | 0.30 | 4.2 | 470 |
| Producer 40 | × | MO | Z3-TO | × | × | MO | 4.30 | 12.7 | 1740 |
| Producer 60 | × | MO | Z3-TO | × | × | MO | 20.8 | 35 | 3810 |
| Producer 80 | × | MO | Z3-TO | × | × | MO | 67.7 | 145 | 6680 |
| Producer 100 | × | MO | Z3-TO | × | × | MO | 172 | 231 | 10350 |

**Table 1.** Detailed experimental evaluation for the set of multi-threaded benchmarks. MO stands for memout; the time spent until memout was in all cases more than 1 hour. U indicates that the termination prover returned "unknown"; Z3-TO indicates a timeout in the Z3 SMT solver.

the experimental results only for Arctor, together with the detailed information about benchmark parameters. The archive with this set of benchmarks is available at `http://www.react.uni-saarland.de/tools/arctor/industrial-benchmarks.zip`

**CUDA.** The *CUDA* benchmark represents an abstraction of the parallel computation of binomial option pricing model on NVidia GPUs, as described here:

`http://docs.nvidia.com/cuda/samples/4_Finance/binomialOptions/doc/binomialOptions.pdf`.

**Make.** The *Make* benchmark models a parallel execution of the make program (achieved by the `make -j N` command), doing a compilation of some program. The model accounts for dependencies between targets and for the workload restrictions.

**Map-Reduce.** The *Map-Reduce* benchmark is a model of Google's implementation of the Map-Reduce framework for its *App Engine* distributed computation platform, as described here: `https://developers.google.com/appengine/docs/java/dataprocessing/`.

Comparison of process parameters, such as number of transitions and instructions per process, shows that Arctor is rather insensitive to them. On the other hand, the behavior of Arctor on the last benchmark demonstrates its current limitation: it does not handle very well processes with high branching degree, due to the resulting combinatorial explosion in the number of traces.

| | Avg. trans. | Avg. instr. | Time(s) | Mem.(MB) | Vertices |
|---|---|---|---|---|---|
| CUDA 2 | 22 | 18 | 0.04 | 3.3 | 86 |
| CUDA 3 | 22 | 18 | 0.09 | 3.7 | 129 |
| CUDA 4 | 22 | 18 | 0.15 | 4.3 | 172 |
| CUDA 5 | 22 | 18 | 0.24 | 4.5 | 215 |
| CUDA 6 | 22 | 18 | 0.33 | 4.5 | 258 |
| CUDA 7 | 22 | 18 | 0.45 | 4.6 | 301 |
| CUDA 8 | 22 | 18 | 0.58 | 5.5 | 344 |
| CUDA 9 | 22 | 18 | 0.72 | 5.5 | 387 |
| CUDA 10 | 22 | 18 | 0.88 | 5.5 | 430 |
| Make 2 | 30 | 54 | 0.04 | 3.6 | 126 |
| Make 3 | 30 | 54 | 0.10 | 4.3 | 189 |
| Make 4 | 30 | 54 | 0.17 | 4.5 | 252 |
| Make 5 | 30 | 54 | 0.26 | 4.5 | 315 |
| Make 6 | 30 | 54 | 0.36 | 4.5 | 378 |
| Make 7 | 30 | 54 | 0.48 | 4.5 | 441 |
| Make 8 | 30 | 54 | 0.62 | 4.6 | 504 |
| Make 9 | 30 | 54 | 0.79 | 5.5 | 567 |
| Make 10 | 30 | 54 | 0.97 | 5.5 | 630 |
| Map-Reduce 2 | 10 | 13 | 0.42 | 4.5 | 238 |
| Map-Reduce 3 | 10 | 13 | 2.50 | 4.5 | 393 |
| Map-Reduce 4 | 10 | 13 | 8.22 | 5.5 | 547 |
| Map-Reduce 5 | 10 | 13 | 31.3 | 6.5 | 767 |
| Map-Reduce 6 | 10 | 13 | 78.7 | 6.5 | 986 |
| Map-Reduce 7 | 10 | 13 | 219 | 7.3 | 1271 |
| Map-Reduce 8 | 10 | 13 | 457 | 8.3 | 1555 |
| Map-Reduce 9 | 10 | 13 | 1053 | 9.3 | 1905 |
| Map-Reduce 10 | 10 | 13 | 1924 | 11.4 | 2254 |

**Table 2.** Experimental evaluation of the Arctor termination prover for the set of industrial multi-threaded benchmarks. Second and third columns represent average numbers of transitions and instructions in a thread of the given benchmark (one transition may comprise several instructions).